NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

OBSTACLE AVOIDANCE CONTROL FOR THE REMUS AUTONOMOUS UNDERWATER VEHICLE

by

Christopher D. Chuhran

September 2003

Thesis Advisor: Anthony Healey

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Obstacle Avoidance Control in the Vertical Plane for the REMUS Autonomous Underwater Vehicle			**5. FUNDING NUMBERS**
6. AUTHOR(S) Chris Chuhran			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT ***(maximum 200 words)***

As the Navy continues its development of unmanned underwater vehicles, the need for total autonomous missions grows. Autonomous Underwater Vehicles (AUV) allow for advances in mine warfare, harbor reconnaissance, undersea warfare and more. Information can be collected from AUVs and downloaded into a ship or battle group's network. As AUVs are developed it is clear forward-look sonar will be required to be able to detect obstacles in front of its search path. Common obstacles in the littoral environment include reefs and seawalls which an AUV will need to rise above to pass. This thesis examines the behavior and control system required for an AUV to maneuver over an obstacle in the vertical plane. Hydrodynamic modeling of a REMUS vehicle enables a series of equations of motion to be developed to be used in conjunction with a sliding mode controller to control the elevation of the AUV. A two-dimensional, 24^{o} vertical scan forward look sonar with a range of 100 m is modeled for obstacle detection. Sonar mappings from geographic range-bearing coordinates are developed for use in MATLAB simulations. The sonar "image" of the vertical obstacle allows for an increasing altitude command that forces the AUV to pass safely over the obstacles at a reasonable rate of ascent and pitch angle. Once the AUV has passed over the obstacle, the vehicle returns to its regular search altitude. This controller is simulated over different types of obstacles.

14. SUBJECT TERMS autonomous, auv, obstacle avoidance, sliding mode controller, hydrodynamic coefficients, REMUS			**15. NUMBER OF PAGES** 63
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

OBSTACLE AVOIDANCE CONTROL FOR THE REMUS AUTONOMOUS UNDERWATER VEHICLE

Christopher D. Chuhran
Lieutenant, United States Navy
B.S., University of Washington, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: Christopher D. Chuhran

Approved by: Anthony J. Healey
Thesis Advisor

Anthony J. Healey
Chairman, Department of Mechanical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

As the Navy continues its development of unmanned underwater vehicles, the need for total autonomous missions grows. Autonomous Underwater Vehicles (AUV) allow for advances in mine warfare, harbor reconnaissance, undersea warfare and more. Information can be collected from AUVs and downloaded into a ship or battle group's network. As AUVs are developed it is clear forward-look sonar will be required to be able to detect obstacles in front of its search path. Common obstacles in the littoral environment include reefs and seawalls which an AUV will need to rise above to pass. This thesis examines the behavior and control system required for an AUV to maneuver over an obstacle in the vertical plane. Hydrodynamic modeling of a REMUS vehicle enables a series of equations of motion to be developed to be used in conjunction with a sliding mode controller to control the elevation of the AUV. A two-dimensional, 24^{o} vertical scan forward look sonar with a range of 100 m is modeled for obstacle detection. Sonar mappings from geographic range-bearing coordinates are developed for use in MATLAB simulations. The sonar "image" of the vertical obstacle allows for an increasing altitude command that forces the AUV to pass safely over the obstacles at a reasonable rate of ascent and pitch angle. Once the AUV has passed over the obstacle, the vehicle returns to its regular search altitude. This controller is simulated over different types of obstacles.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Professor Anthony J. Healey, for his expert insight, direction, and assistance during the development of this work.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

As the twenty first century begins the underwater world is becoming increasingly important to both civilian and military matters. The decline of precious natural resources and the desire to understand the natural world and how it changes motivates entire fields of scientific underwater research. In the military, the Chief of Naval Operations' new naval doctrine, Sea Power 21, calls for a far-reaching collection of information that can be used by battlefield commanders to carry out their duties called ForceNet (Clark, 2002),. Within ForceNet, this underwater world also plays a key factor. Advances in mine warfare and submarine technology as well as increasing importance of littoral control have created a vast need for underwater supremacy and underwater awareness.

Unmanned Underwater Vehicles (UUVs) are increasingly being found to be more than capable to satisfy the requirements of these civilian and military organizations (see Stutz, 2003 for current military application). UUVs are small submersible vehicles that contain independent propulsion systems and are capable of carrying sensors such as side-scan sonar, video cameras and an assortment of oceanographic measuring devices. UUVs are highly desirable as they can take away or at least limit the level of human risk and human involvement in a mission. A UUV can accomplish longer missions without risk of fatigue or marine animal attack. Furthermore, a UUV can be highly stealthy and can have a high capacity for data storage.

One type of UUV is an autonomous underwater vehicle (AUV). An AUV operates completely independent of human control. There are no tethers connected to an AUV, as compared to a remote operated vehicle (ROV). An AUV can therefore travel further distances away from its home base. Advances in acoustic underwater communication allow data to be relayed back and forth from the AUV to a home base allowing an AUV to contribute to a real time ForceNet type of data collection. Onboard computers can also store data to be downloaded at a later time. This data can be used to identify important oceanographic characteristics of a body of water for example or to map out a mine field for future littoral operations.

The future of AUV operations is full of possibilities, but technology still needs to be developed for an AUV to completely mission capable. One system currently in development by the Woods Hole Oceanographic Institution, named REMUS (Remote Environmental Measuring Unit) is a single propeller, 5m long vehicle (von Alt, 1994). The REMUS is currently configured with an altimeter and a side scan sonar. Its lithium-ion battery allows for up to 24 hours of mission life. However, studies indicate a forward look capability will be required to enable obstacle avoidance along its path. A forward-look sonar designed to a size that would fit a small (approximately 5 m long) AUV would satisfy this requirement, but is still relatively new and untested. With a successful forward-look sonar and AUV would be able to detect obstacles in front of it and maneuver to avoid collision.

B. MOTIVATION

Study in the field of underwater robotic AUVs has been done since 1960 and experimental prototypes were available in the 1980's. More history on the development of AUV's can be found in (Blidberg, 2001). AUV's are capable of operating in numerous underwater environments, including littorals and even under polar icecaps. The have the capability to search, detect and classify objects using its side-scan sonar and video camera and they can also measure oceanographic data such temperature, salinity and current. These capabilities are necessary for both oceanographic research and military operations such as mine hunting and harbor reconnaissance. At this time however, most AUVs travel on a fixed path through the water where a certain level of knowledge of the seafloor is known. A problem exists when AUVs travel into unknown waters or where the local bathymetry is not predetermined. On this occasion and AUV is highly likely to come into contact with underwater obstacles such as coral reefs, sea walls and shipwrecks. Obstacle avoidance technology, that is the creation of computer algorithms that will determine maneuvering options for an AUV confronting an obstacle, are still in development. In a sometimes chaotic and treacherous underwater environment it can not be hoped to be able to plan for every contingency, particularly in military operations on hostile littorals. It is therefore necessary to develop a logic system within the AUV that will allow it to recognize an obstacle and make a correct maneuver to avoid

the obstacle and then return to its mission path as soon as possible. The complexities of a three dimensional environment make obstacle avoidance algorithms difficult to develop. Underwater, an AUV has six degrees of freedom and all six degrees can be affected by the simplest of turns or dives. Therefore this thesis selects to investigate motion in the vertical plane only. The object of this thesis is to develop a model of an AUV that is capable of recognizing an obstacle that is must ascend to avoid and then maneuver correctly and return to its flight path as soon as possible.

C. OBSTACLE AVOIDANCE FOR AUTONOMOUS UNDERWATER VEHICLES

Consideration for obstacle avoidance techniques involving AUVs has gone on for quite some time. Most of the work involves avoiding obstacles in the horizontal plane (for examples see Fodrea, 2002 or Kamon, 1997). In these cases an AUV is following a specific track and obstacles to be avoided result in course deviations to avoid the obstacle and then to return to the original track.

There has been considerable less work involved in avoiding obstacles in the vertical plane. Work in this area is typically called bottom following or bottom navigation. While working with the Autonomous Benthic Explorer (ABE) a method of bottom following was developed using only an altimeter (Singh, et. al, 1995). In this case the altimeter was positioned to be able to read the terrain slightly forward of the vehicle. The controller for ABE was designed to ascend quickly in response to rising terrain and to descend slowly once an obstacle was passed.

Similar work was done while experimenting with the AUV Odyssey (Bennet, et.al, 1995). In this case previous altimeter readings were put into memory and a slope of the bottom was calculated. As the slope of the bottom increased when approaching an obstacle, the vehicle would ascend to avoid the obstacle. If the slope was too great for the vehicle to ascend in time the AUV could also slow down or turn in attempt to avoid the obstacle.

D. SCOPE OF THESIS – THE REMUS VEHICLE

The intent of this thesis is to develop an obstacle avoidance algorithm to be used on a REMUS AUV equipped with a forward-look sonar. The REMUS is designed to perform hydrographic analysis in the Very Shallow Water (VSW) zone from 40 to 100 ft deep. Fig. 1 shows a picture of the vehicle that is currently being used at the Naval Postgraduate School in Monterey, CA. Dimensions of the vehicle are also shown below in Table 2.

Figure 1. REMUS Vehicle

Table 1 REMUS Functional and Physical Characteristics

PHYSICAL/FUNCTIONAL AREA	CHARACTERISTIC
Vehicle Diameter	7.5 in
Vehicle Length	62 in
Weight in Air	80 lbs
External Ballast Weight	2.2 lbs
Operating Depth Range	10 ft to 66 ft
Transit Depth Limits	328 ft
Typical Search Area	875 yds X 1093 yds
Typical Transponder Range	1640 yds
Operational Temperature Range	+32F to +100F
Speed Range	0.5 knots to 5.6 knots
Maximum Operating Water Current	2 knots
Maximum Operating Sea State	Sea State 2
Battery	1 kW-hr internally rechargeable Lithium-ion
Endurance	20 hours at 3 knots; 9 hours at 5 knots

The REMUS AUV is not currently equipped with a forward-look sonar. It does have two side scan sonar that are capable of detecting objects underneath and to the sides of the REMUS. NPS is currently in the process of implementing a forward-look blazed array sonar into the nose section. A "blazed array" sonar refers to a sonar with elements that are rotated out of plane from each other (Thompson, 2001).

In creating this obstacle avoidance algorithm, the thesis will not attempt to try model realistic sonar data. This data typically requires filtering due to noise and back scatter inherent in sonar technology. Additionally, sensors may fail or partially fail and this will not be considered either. This algorithm will assume that some level of filtering has occurred and that an object has been detected.

This algorithm will specifically deal with two different types of obstacles. One obstacle will be an abrupt ramp that rises 6m above the ocean floor and then maintains that elevation for the rest of the transit. A second obstacle will be an abrupt rise that quickly drops off back to its initial ocean floor elevation. These two obstacles will allow the algorithm to display its ability to command the vehicle to rise above an obstacle and then either maintain a new depth for the ramp or return to its original depth for the abrupt rise. It should be pointed out that when following altitude commands, altitude from a downward look acoustic system is used as the feedback signal to a depth controller, while water depth from a pressure cell is measured but not used for control.

E. THESIS STRUCTURE

The development of an obstacle avoidance algorithm for the REMUS vehicle will require a series of procedures. First, the dynamics of the REMUS vehicle must be developed to understand how the REMUS reacts to its underwater environment and it propeller/control surface system. Equations of motion will be developed to account for the degrees of freedom involved in a 2-D vertical plane. These equations of motions will require the development of REMUS' hydrodynamic coefficients. Secondly, a robust altitude controller will developed to ensure the REMUS can safely manipulate its altitude to both maintain a desired search altitude and to ascend or descend to avoid obstacles. Finally, the obstacle avoidance algorithm will be developed implementing the REMUS'

equations of motion and the altitude controller. The algorithm will allow the REMUS to safely avoid both the abrupt ramp and the abrupt rise obstacles.

Chapter II will begin this development with a discussion of the development of the equations of motion and hydrodynamic coefficients. Chapter III will describe the altitude controller and implement the equations of motion into this controller. Chapter IV will describe the obstacle avoidance algorithm that has been developed. Chapter V will discuss results of the obstacle avoidance algorithm used in a simulated MATLAB environment. Finally, Chapter VI will discuss the overall results of this thesis and suggest some future work that can be done in this area.

II. DYNAMIC MODELING

A. GENERAL

The first step in designing a control system for a rigid body is to understand the dynamics. An underwater rigid body in free space with out restraints is considered as a model. This model will have six degrees of freedom and the equations of motion will be derived using a Newton-Euler approach (Healey, 1995). An underwater vehicle has forces acting on it from its propulsion system and from the surround hydrodynamic and hydrostatic effects of the water. The interaction of the water and the vehicles body causes forces of lift and drag. These forces can analyzed and used to develop controlling equations used to determine how to maneuver the vehicle using its propulsive force and control surfaces.

B. EQUATIONS OF MOTION IN THE VERTICAL PLANE

1. Introduction

This section explains the derivation of the equations of motion used to develop the vertical plane controller. These equations were adapted from the work on the ARIES AUV (Healey Notes) and uses the following assumptions:

- the vehicle behaves as a rigid body;
- the earth's rotation is negligible as far as acceleration components of the vehicle's center of mass is concerned
- the primary forces that act on the vehicle have inertial and gravitational origins as well as hydrostatic, propulsion, thruster and hydrodynamic forces from lift and drag.

2. Coordinate System

The underwater coordinate system has both a global reference frame, which encompasses the ocean environment and a local coordinate system, which is centered upon the rigid body and moves with the rigid body. The global reference frame is defined as OXYZ, with the origin at O, and the directions North (X), East (Y) and Down

(Z) oriented as a right hand set with unit vectors **I, J, K**. Therefore, a vehicle will have position $\mathbf{R_o}$ such that

$$\mathbf{R_o} = [X_o\mathbf{I} + Y_o\mathbf{J} + Z_o\mathbf{K}]. \quad (1)$$

The local coordinate from is defined o,x,y,z with an origin at o and the positions forward (x), starboard (y) and down (z) as a right hand set with unit vectors **i, j, k.** This local coordinate frame locates any component on the vehicle with respect to an origin that is typically located on the longitudinal axis near the center of mass. An example of the global and local coordinate systems can be seen below in Fig. 2.

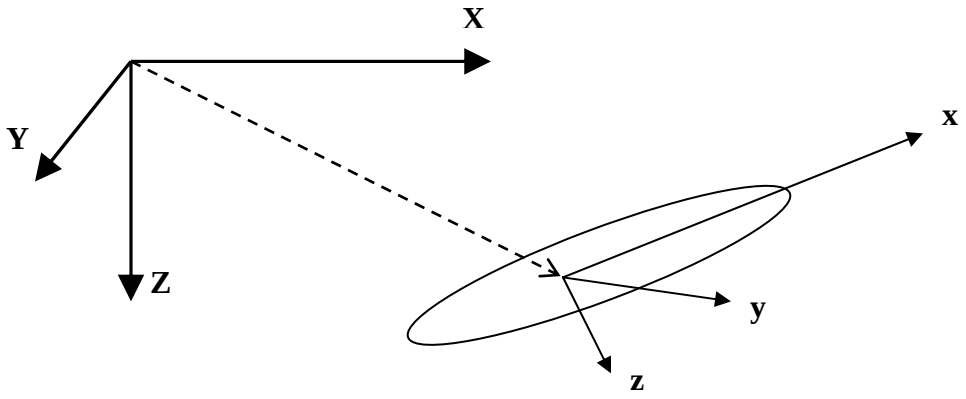

Figure 2. Global and Local Coordinate System (Healey)

It should be noted that the local origin on the vehicle does correspond to either the center of mass or center of buoyancy. Depending upon payload configuration, the center of mass can shift, typically only along the longitudinal axis. The locations of the centers of mass (ρ_g) and buoyancy (ρ_b) are very important and will be used later when developing the equations of motion.

3. Angular Position

It is necessary to also be able to define the attitude of and underwater vehicle with respect to the global reference frame. This "attitude" may be necessary when trying to control the orientation of the vehicle with respect to another object. The rates of change

of these attitudes are also important as the resulting changes in centers of mass velocity are important to the dynamic equations of motion.

Three Euler angles are used to describe this attitude. The azimuth rotation, ψ, is a positive (right hand rule) rotation about the global **Z** axis. Next a rotation θ (right hand rule positive up) about the new **Y** axis is defined. Finally there is a positive rotation φ about the new **X** axis. These three angles will always describe the attitude of the vehicle.

Healey (notes) then shows how a transformational matrix (T (φ,θ,ψ)) can be formed that can transfer coordinates to and from local and global coordinates. The transformational matrix is repeated below:

$$T(\varphi,\theta,\psi)=\begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi-\sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi+\cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \quad (2)$$

4. Kinematics

A velocity vector is now defined in both local and global coordinates. Globally, an object clearly can move at a certain velocity in the reference frame **OXYZ**. This may be how an object is measured by radar or sonar. These speeds are denoted as $\dot{X}, \dot{Y}$ and $\dot{Z}$. However, an underwater vehicle has means of measuring its own speed, sometimes without the reference of an accurate global position. The local velocity vector $[u,v,w]^{-1}$ includes the surge speed (u), the side slip or sway (v) and the heave velocity (w). The local and global velocities can be transformed from one another using the transformation matrix (T):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = T(\phi,\theta,\psi)\bullet\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad (3)$$

Finally, a relationship between the angular rate of change can also be determined. Although the Euler angles rate of change is most important for the equations of motion, these values cannot commonly be calculated by any device. However, the vehicle does have sensors that can calculate its own rates of change with respect to the local

coordinate system. These angle rates can then be transformed to the Eulerian angle rates. The local angles are defined by the vector $\boldsymbol{\omega} = [p,q,r]^T$ in which p is roll rate, q is pitch rate and r is yaw rate. The relationship is:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \sum T(\phi)\cdot T(\theta)\cdot T(\psi)\cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + T(\phi)\cdot T(\theta)\cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + T(\phi)\cdot \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

with the result:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5}$$

It can be seen for smaller angles that:

$$\dot{\phi} = p;\ \ \dot{\theta} = q;\ \ \dot{\psi} = r. \tag{6}$$

5. Equations of Motion

With the above definitions of the coordinate frame, angular position and kinematics of the vehicle now described, the equations of motion (EOM) can now be developed. The six EOM are developed from two equations, the sum of forces acting upon the rigid body and the sum of moments acting upon the rigid body. The sum of forces acting upon the rigid body, commonly developed in most dynamic theory is:

$$F = m\left(\dot{v} + \dot{\omega}\times\rho_g + \omega\times\omega\times\rho_g + \omega\times v\right) \tag{7}$$

The equation of the sum of moments acting upon the rigid body is derived from equating the sum of the applied moments about the vehicle's center of mass to the rate of change of angular momentum of the vehicle about its center of mass. The resulting equation of motion is:

$$M_o = I_o\dot{\omega} + \omega\times\left(I_o\omega\right) + m\left(\rho_g\times\dot{v} + \rho_g\times\omega\times v\right) \tag{8}$$

With the addition of weight and buoyancy terms that act at the centers of gravity and mass, Healey (notes) derives the EOM for a six degree of freedom model as:

SURGE EQUATION OF MOTION

$$m[\dot{u}_r - v_r r + w_r q - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] + (W - B)\sin\theta = X_f \quad (9)$$

SWAY EQUATION OF MOTION

$$m[\dot{v}_r + u_r r - w_r p + x_G(pq + \dot{r}) - y_G(p^2 + r^2) + z_G(qr - \dot{p})] - (W - B)\cos\theta\sin\phi = Y_f \quad (10)$$

HEAVE EQUATION OF MOTION

$$m[\dot{w}_r - u_r q + v_r p + x_G(pr - \dot{q}) + y_G(qr + \dot{p}) - z_G(p^2 + q^2)] + (W - B)\cos\theta\cos\phi = Z_f \quad (11)$$

ROLL EQUATION OF MOTION

$$I_x\dot{p} + (I_z - I_y)qr + I_{xy}(pr - \dot{q}) - I_{yz}(q^2 - r^2) - I_{xz}(pq + \dot{r}) + m[y_G(\dot{w} - u_r q + v_r p) \quad (12)$$

$$- z_G(\dot{v}_r + u_r r - w_r p)] - (y_G W - y_B B)\cos\theta\cos\phi + (z_G W - z_B B)\cos\theta\sin\phi = K_f$$

PITCH EQUATION OF MOTION

$$I_y\dot{q} + (I_z - I_z)pr - I_{xy}(qr + \dot{p}) + I_{yz}(pq - \dot{r}) + I_{xz}(p^2 - r^2) - m[x_G(\dot{w} - u_r q + v_r p) \quad (13)$$

$$- z_G(\dot{u}_r - v_r r + w_r q)] + (x_G W - x_B B)\cos\theta\cos\phi + (z_G W - z_B B)\sin\theta = M_f$$

YAW EQUATION OF MOTION

$$I_z\dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) + m[x_G(\dot{v}_r + u_r r - w_r p) \quad (14)$$

$$- y_G(\dot{u}_r - v_r r + w_r q)] - (x_G W - x_B B)\cos\theta\sin\phi - (y_G W - y_B B)\sin\theta = N_f$$

Where:

W = weight
B = Buoyancy
I = mass moment of inertia terms
u_r, v_r, w_r = component velocities for a body fixed system with respect to the water
p, q, r = component angular velocities for a body fixed system
x_B, y_B, z_B = position difference between geometric center and center of buoyancy
x_G, y_G, z_G = position difference between geometric center and center of gravity
X_f, Y_f, Z_f, K_F, M_f, N_f = sums of all external forces acting in the particular body fixed direction

For a vehicle operating in the vertical plane the following assumptions can be made:

$$v = 0,\ p = 0,\ r = 0,\ \dot{v} = 0,\ \dot{p} = 0,\ \dot{q} = 0\ .$$

Furthermore, by inspection of the REMUS vehicle data collected by Prestero, the center of mass is located below the origin and the center of buoyancy is located at the origin of the local coordinate axis. Therefore:

$x_G = 0$, $y_G = 0$, z_G =1.96 e-2 m, and $x_B = 0$, $y_B = 0$, $z_B = 0$.

The simplified equations of motion are therefore:

$$m\dot{w}_r - mU_o q - (W - B) = Z_f(t) \tag{15}$$

$$I_y \dot{q} + z_G W \sin\theta = M_f(t) \tag{16}$$

$$\dot{\theta} = q\cos\phi = q \tag{17}$$

$$\dot{Z} = -U_o \sin\theta + w_r \cos\theta\cos\phi \tag{18}$$

6. Hydrodynamic Coefficients

In addition to the forces of inertia and propulsion upon the vehicle, the surrounding water creates an "added mass" or an additional force that must be accounted for in the equations of motion. These added mass force arise due to the pushing of water as the vehicle travels through the water. These forces increase and decrease depending upon the vehicles angle of attack and side slip. The added mass forces create both a heave force in the vertical plane, Z_f and a pitching moment M_f. As described by Healey (notes) the added mass functions, pertinent to the vertical plane, can be described as:

$$\Delta Z_f = f(w_r, \dot{w}_r, q, \dot{q}, t) \tag{19}$$

$$\Delta M_f = f(w_r, \dot{w}_r, q, \dot{q}, t) \tag{20}$$

The added mass forces can be linearized using Taylor series expansion terms in individual motion components. These expansion terms are called 'hydrodynamic coefficients' and are determined by the shape of the vehicle. These values are typically arrived by experimental data. The values used for this dynamic model came from

Prestero's work on the REMUS vehicle (Prestero, 2001). The equations used to determine the heave force and pitching moment due to added mass (Healey 2001) are:

$$Z_f = Z_{w_r} w_r + Z_{\dot{w}_r} \dot{w}_r + Z_q q + Z_{\dot{q}} \dot{q} \tag{21}$$

$$M_f = M_{w_r} w_r + M_{\dot{w}_r} \dot{w}_r + M_q q + M_{\dot{q}} \dot{q} \tag{22}$$

Where:

Z_{w_r} = coefficient of heave force induced by angle of attack
$Z_{\dot{w}_r}$ = coefficient for added mass in heave
Z_q = coefficient of heave force induced by angle of attack
$Z_{\dot{q}}$ = coefficient for added mass in pitch
M_{w_r} = coefficient of pitch moment from heave
$M_{\dot{w}_r}$ = coefficient for added mass moment of inertia in heave
M_q = coefficient of pitch moment from pitch
$M_{\dot{q}}$ = coefficient of pitch moment from pitch

The values used for these coefficients were unaltered from Prestero's work. Table 2 below shows the values used.

Table 2 REMUS Hydrodynamic Coefficients

Z_{w_r}	-6.66 e 1 kg/s
$Z_{\dot{w}_r}$	-3.55 e 5 kg
Z_q	-9.67 e 0 kg m/s
$Z_{\dot{q}}$	-1.93 e 0 kg m
M_{w_r}	+3.07 e 1 kg m/s
$M_{\dot{w}_r}$	-1.93 e 0 kg m
M_q	-6.87 e 0 kg m^2/s
$M_{\dot{q}}$	-4.88 e 0 kg m^2

7. Control Surface

Additionally, the REMUS fin must be accounted for, as it controls the vertical movement of the vehicle. Johnson (2001) showed that fin action produces forces that when linearized are $Z_\delta\delta(t)$ and $M_\delta\delta(t)$. The final equations of motion are therefore:

$$m\dot{w}_r - mU_o q - (W - B) = Z_{w_r} w_r + Z_{\dot{w}_r} \dot{w}_r + Z_q q + Z_{\dot{q}} \dot{q} + Z_\delta \delta_r(t) \tag{23}$$

$$I_y \dot{q} + z_G W \sin\theta = M_{w_r} w_r + M_{\dot{w}_r} \dot{w}_r + M_q q + M_{\dot{q}} \dot{q} + M_\delta \delta_r(t) \tag{24}$$

$$\dot{\theta} = q \cos\phi = q \tag{25}$$

$$\dot{Z} = -U_o \sin\theta + w_r \cos\theta \cos\phi \tag{26}$$

8. Matrix Form

These final equations of motion, (23) through (26), can be placed into matrix form which allows for manipulation with MATLAB. The matrix form follows the standard control law format of $M\dot{x} = Ax + Bu$, as shown below:

$$\begin{bmatrix} (m - Z_{\dot{w}_r}) & -Z_{\dot{q}} & 0 & 0 \\ -M_{\dot{w}_r} & I_y - M_{\dot{q}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} Z_{w_r} & (Z_q + mU) & 0 & 0 \\ M_{w_r} & M_q & -z_G & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -U & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ Z \end{bmatrix} + \begin{bmatrix} Z_\delta \\ M_\delta \\ 0 \\ 0 \end{bmatrix} \delta(t) = \begin{bmatrix} (W - B) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{27}$$

III. CONTROL METHODS AND ARCHITECTURE

A. GENERAL CONTROL THEORY

An underwater vehicle operates with six degrees of freedom and must respond to influences of hydrostatic and hydrodynamic forces from the ever changing environment of the ocean. Additionally, an AUV must respond to obstacles within the ocean environment such as the changes in the sea floor depth, mines, shipwrecks, reefs, etc. Finally, autonomous control allows for no human interface while the AUV conducts its mission, therefore all aspects of control must be determined before the mission starts.

An AUV is aided by sensors which can measure the vehicles position, speed, altitude and also the rate of change of position, speed and altitude. Additionally, a forward-mounted sonar can detect obstacles in the AUV's path. Therefore, the challenge is to develop a planned path and then create a controller that will execute this planned path. This planned path can not hope to account for every obstacle or other environmental issue, therefore the controller must be able to deviate from the planned path to avoid obstacles and then return to the planned path when the obstacle has passed. The controller has the benefit of input from the AUV's navigational system, including an altimeter, forward-mounted sonar and a speed sensor.

Feedback controllers are required with AUVs to provide autopilot functions. Work done on the ARIES vehicle (Healey and Marco, 2001) for example, among many others, has proven that feedback controllers can properly maintain depth and track during an AUV mission. Feedback controllers must be robust enough to account for changes in ocean current and changes in ocean floor depth. To control the highly responsive REMUS, for this work, a Sliding Mode Controller (SMC) has been developed and used. The SMC can be a robust controller and adaptive to the underwater environment. Additionally, it is a robust method with a theory which allows n^{th} order systems to be effectively replaced by a (n-1) order system and the ability to "tune" this controller with only a few disposable parameters.

The SMC controller is effective by using the feedback of specific motion variables, obtained from the AUV's sensors, to drive the vehicle's actuators (rudders and elevators). For the purpose of this thesis, only control in the vertical plane is necessary. Therefore only the altitude controller is used, implementing inputs of vehicle motion state w_r, q and θ and altitude above bottom, to determine the elevator angle, δ, required. It is assumed that the REMUS maintains a constant horizontal speed, U_o, during its entire mission run. Additionally, any changes in the horizontal plane such as yaw or roll caused by diving are neglected.

B. SLIDING MODE CONTROL

A multivariable sliding mode controller is used to provide accurate altitude control of the AUV. The non-linear EOM for the REMUS vehicle in the vertical plane were linearized in the previous chapter to allow for this SMC. To create the SMC, the general form of the equations of motion is used:

$$x = A\dot{x} + Bu \tag{28}$$

where $x \in \mathbf{R}^{n*1}; A \in \mathbf{R}^{n*n}; B \in \mathbf{R}^{n*r}; u \in \mathbf{R}^{r*1}$, and u is the elevator angle. A sliding surface, σ is then created, in which $\sigma = 0,\ \sigma \in \mathbf{R}^{\rho*1}$. The sliding surface is defined as:

$$\sigma = s'\tilde{x};\ \ \tilde{x} = x - x_{com} \tag{29}$$

where **s**' is a vector of directions in the state error space. As discussed in SMC theory, the controller works by driving the sliding surface to zero, using the requirement that:

$$\dot{\sigma}\sigma < 0\ \forall t. \tag{30}$$

As the sliding surface approaches zero the error, $\tilde{x}$, between the state variables (x) and the command (x_{com}) is zero. By definition of the sliding mode controller, the system dynamics must exhibit stable sliding on the surface when σ = 0. Therefore, **s'** can be determined by observing that the closed loop dynamics are given by the poles of the closed loop matrix as,

$$(A - bk_2) = A_c,\ with\ k_2 = [s'B]^{-1}s'A \tag{31}$$

where k_2 is chosen by pole placement and $A_c s' = 0$ to achieve the condition $\sigma = 0$ The linear sate feedback gains for each state used are found by using the eigenvectors of the A_c matrix. The sliding surface is then as follows:

$$\sigma(t) = -s_1 q(t) - s_0 w_r(t) + s_2(\theta_{com} - \theta(t)) + s_3(Z_{com} - Z(t)) \quad (32)$$

The poles selected for the REMUS model SMC simulation were based on trial and error. A desirable system response was found with poles at [0, -0.6, -0.62 0.63]. In order for the single sliding constraint for the single input system, implied by $\sigma = 0$, a pole must be placed at the origin. The remaining three poles are all in the left hand plane, required for stable dynamics. The gains obtained from this pole placement, using the MATLAB "place" command, were $[k_1\ k_2\ k_3\ k_4]$ = [-1.5710 0.3131 0.1888 0] for [w_r q Z θ] respectively. Using the gains determined above and the sliding surface equation (30), the commanded elevator plane in the controller becomes:

$$\delta(t) = -k*x-\eta*\tanh((\sigma/\varphi)) \quad (33)$$

where η and φ are tuning factors.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. OBSTACLE AVOIDANCE MODEL

A. NORMAL AUV TRAVEL

This thesis considers the travel of the REMUS in a 2 dimensional environment. This environment consists of the vertical plane which allows for tracking of the vehicle's depth and its forward progress in the water. It is assumed that the vehicle travels at a constant speed of 1.5 m/s or about 3 knots, which is the normal search speed for the REMUS vehicle. During a typical mission, the sliding mode controller will constantly maintain a commanded altitude. This controller will allow for gradual changes in the depth of the water. The altitude control is enabled by the REMUS' altimeter and the typical search altitude using the RDI Doppler navigator set point is 3m above the ocean floor. As the ocean floor rises or descends, feedback from the altimeter will create and error signal with the commanded altitude. This altitude error will be corrected by actuation of the REMUS elevator planes. For gradual changes in ocean floor depth the correction is near instantaneous and the elevator planes would return to zero degrees assuming no other factors are involved. The REMUS does have other factors that affect its altitude control however. The REMUS is not always a neutrally buoyant vehicle. This will depend upon payload of course and for purposes of this thesis it is assumed that the vehicle is positively buoyant by a weight of 7 N. This positive buoyancy provides a constant upwards force that must be countered by the vehicles elevator planes, therefore the elevator planes do not return to a steady state of zero degrees on a level ocean floor. Furthermore, the altitude controller also can interact with ocean current. Although not utilized in this thesis, Healey (Healey, 1995) has shown how current affects can be implemented into the controller to maintain the proper altitude at all times.

B. FORWARD LOOK SONAR

Although the vehicle's altitude can be maintained for minor changes in ocean floor depth, buoyancy effects and ocean current, this controller is not adequate for abrupt changes in ocean floor depth typically caused by reefs, seawalls or other obstructions. A altimeter, for example, would not observe a 3 m tall coral reel until the REMUS had already crashed into it! Even with some forewarning of a large obstacle in front, the

REMUS requires considerable to notice to advance at a reasonable rate of ascent and pitch angle. To account for this deficiency, forward-look sonar is required. A typical forward-look sonar with 400 KHz pings can have ranges of over 100m and a 24^{o} vertical scan as shown below in Fig. 3.

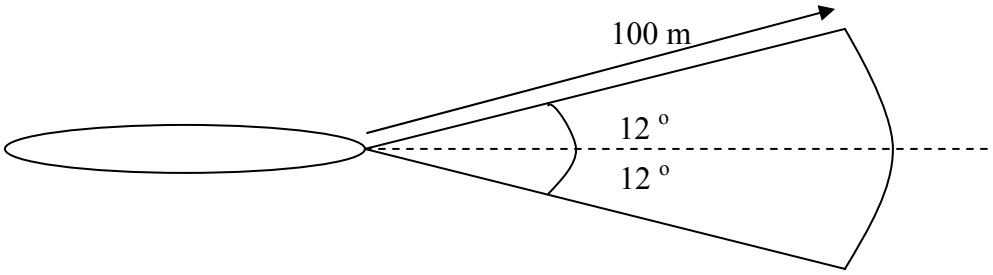

Figure 3. Forward Look Sonar Model

With forward-look sonar installed on a REMUS vehicle, coral reefs and other such obstacles can now be detected at a distance adequate enough to allow for a gradual ascent over the object. Small forward-look sonars are in a relatively recent phase of development. Work currently being done on a blazed array sonar (Thompson 2001) is being considered for use in experiments at the Naval Postgraduate School's AUV program. This array has 450 $\pm$ 150 KHz, 25 ^{o} vertical beam pattern with 1^{o} individual beams for image resolution.

A final problem does exist due to the geometry of the REMUS and the abilities of the altimeter and the forward-look sonar. Looking below at Fig. 4, there is a "blind spot" located below the lowermost scan of the forward-look sonar and the altimeter. Clearly, the forward-look sonar will detect the object as it moves forward, but this area can cause problems with the controller when trying to determine how to maneuver over the object and when the object has safely passed. This will be seen in the chapter concerning vehicle simulation.

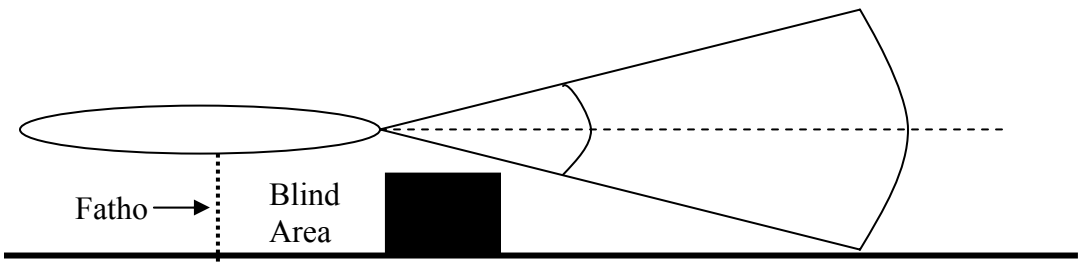

Figure 4. Depicting "Blind Area" between Forward-Look Sonar and Fatho

C. OBSTACLE AVOIDANCE

The approach in this thesis for obstacle avoidance in the vertical plane makes the following assumptions:

- All obstacles must be avoided vertically
- All obstacles are seen clearly by the forward look sonar
- There is no ocean current
- There is no translational or rotational motion in the horizontal plane

With these assumptions the following process can occur when the REMUS approaches an obstacle blocking it path. In the two dimensional realm, the REMUS will detect the obstacle from the forward-look sonar and receive a series of bearings and ranges to that obstacle. After a successful sweep of the obstacle the REMUS' onboard computer system should be able to determine (1) the height of the object and (2) the distance to the object. The purpose of the obstacle avoidance algorithm is to allow the REMUS to detect the obstacle early enough to allow for a gradual rate of ascent and a small pitch angle. The energy saving method will allow the REMUS to stay on station for longer periods of time.

Upon determination of the height and range of the obstacle, the obstacle avoidance algorithm can then plot a "slope" of increasing altitude that will allow the REMUS to pass over the obstacle at a height of 3m. An example path is shown below in Fig. 5.

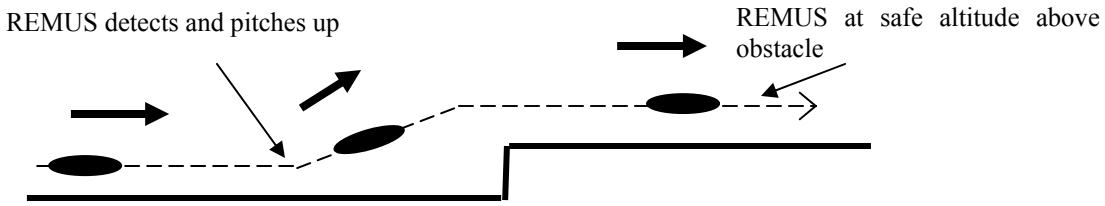

Figure 5. REMUS flight path over obstacle

The obstacle avoidance algorithm works using the principles of "danger bearings" found in navigation. A danger bearing tells a mariner that there is danger if the ship is to one side or another of the danger bearing. Similar to a danger bearing, a zone is created within the forward-look sonar search path. Any object that is detected within this zone is considered to be an obstacle to be avoided. This zone must account for the fact that the

forward-look sonar will receive bottom bounce from the ocean floor. Fig. 6 below depicts the zone in a typical forward-look sonar search path. This zone does not utilize the entire range of typical forward-look sonar. This is due to the fact that the REMUS is a highly maneuverable vehicle and will not require 100 m to rise above obstacles found in its path. Fig. 6 shows that there is a bearing/range in which the length of the zone is constant, set at 45m for this thesis, and there is a second bearing/range where the length decreases as it is actually reading across the ocean floor. Any object that is detected within this zone should trigger the avoidance control algorithm causing the REMUS to rise.

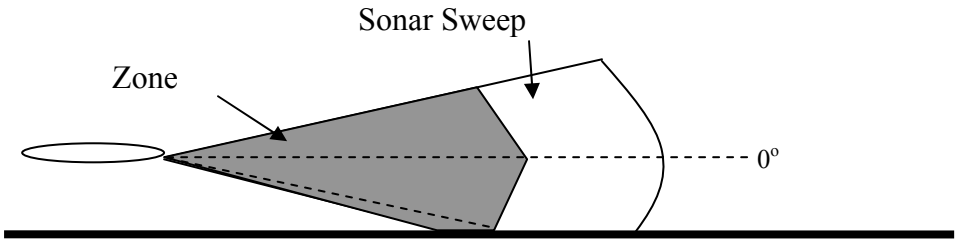

Figure 6. Obstacle Avoidance Zone in Forward-Look Sonar Sweep

As described previously, the REMUS has an altitude controller which uses feedback from the altimeter to maintain a 3m altitude above the ocean floor. When an obstacle is detected within the zone of the sonar sweep the obstacle avoidance algorithm creates a new altitude command that increases linearly as the REMUS approaches the obstacle. The rate of increase or slope of the altitude command is based solely on the height and range of the obstacle. As the REMUS ascends to avoid the obstacle the pitch increases as well.

A problem can arise if the pitch increases too much, as shown in Fig. 7, the obstacle can now no longer be detected by either the forward-look sonar of the altimeter. If the algorithm is simply on on/off controller, each time the obstacle is removed from the sonar's field of view the vehicle will attempt to return to its original altitude. This will create a sinusoidal type flight path which both wastes energy and puts the vehicle in danger of collision.

There are a few different ways to account for the "blind spot" in the forward-look sonar. The algorithm may allow for the REMUS to blindly ascend until the obstacle has

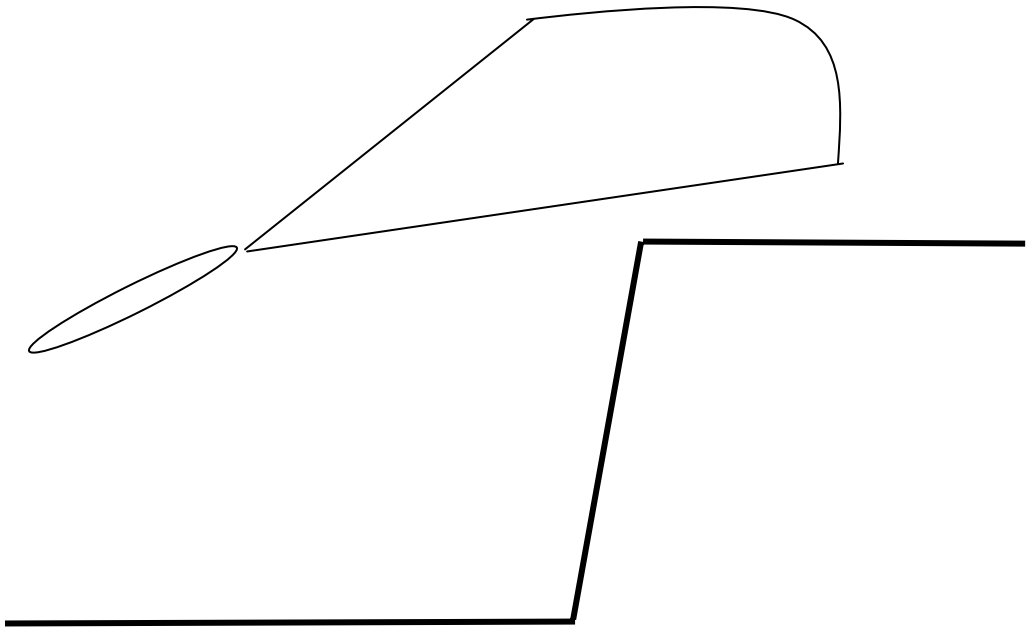

Figure 7. At high pitch angles, REMUS cannot "see" obstacle anymore

passed using a dead-reckoning approach. Based on the knowledge of the bearing and range of the obstacle, a sloping altitude command may be created and the vehicle can be made to maintain that rate of ascent for a given period of time which would equal the estimation of time required to travel to the top of the obstacle. Once reaching the top of the obstacle the vehicle would then look to see if it has successfully passed the obstacle. Although effective, this method has several obvious dangers, particularly if there is a strong or uneven current. Additionally, by not using its sensor during ascent, the REMUS may fail to see obstacles beyond the first obstacle.

The solution used for this thesis is two-fold. First, the REMUS is controlled such that its pitch angle is as shallow as possible. As well as being an efficient energy saving technique, this allows the forward-look sonar the ability to always be able to detect the obstacle in front of it while still looking for new obstacles beyond it. This method is effective in almost all cases, except for extreme cases where an obstacle is not detected quickly enough. In murky waters with plenty of acoustic disturbance this may be a

problem. Secondly, a delay is created within the algorithm so that the REMUS does not immediately attempt to descend once the obstacle is no longer inside the zone. This delay is based somewhat upon the dead-reckoning approach. The vehicle will not begin to descend until it has passed the estimated position of the obstacle; however, the vehicle sensors are still being fed back and if a new obstacle is located closely beyond the first obstacle action will be taken. A block diagram of the object avoidance system dynamics can be seen in Fig. 8 below.

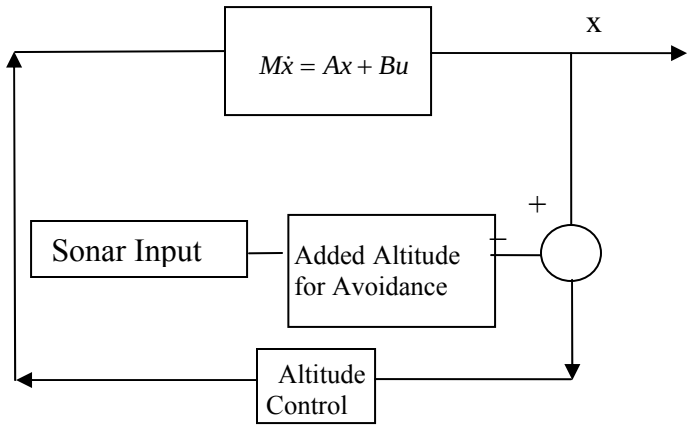

Figure 8. Block Diagram System Dynamics

V. VEHICLE SIMULATION

A. OCEAN ENVIRONMENT MODELING

To test the obstacle avoidance controller a two-dimensional ocean environment was created. To do this, a MATLAB function was created named remusderivalt.m. The ocean environment is created such that the surface of the ocean is the Z reference point and the depth of the ocean increases positively along the Z axis as shown in Fig. 9. The X-axis increases positively in the horizontal direction according to the forward direction of the AUV. For this work, the ocean floor is assumed to have a normal depth of 20m, which changes only for the obstacle to be avoided.

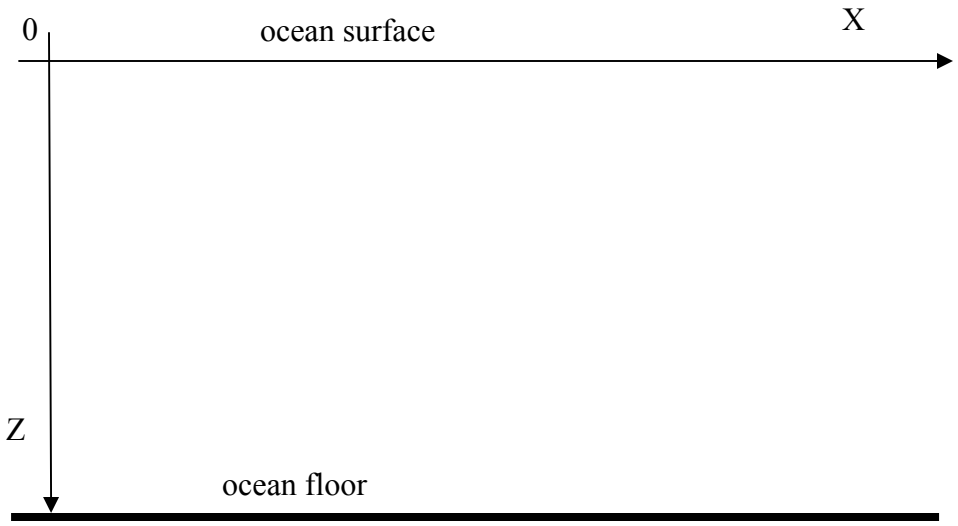

Figure 9. Ocean Floor Model

In MATLAB, the ocean floor is simulated by two arrays, X and Z which simply correspond to the X and Z locations of the ocean floor. The MATLAB function remusderivalt.m consists of two separate X and Z arrays. The first set of arrays, called X_Model and Z_Model represent the ocean floor in a space-domain model. The space domain model is required in the sonar simulation described below. The second array, called Z_t, exists in a time domain and represents the depth of the ocean floor. These arrays are used in the altimeter model also described below.

By using the MATLAB ode23 function, the REMUS equations of motion are integrated over time and the vehicle moves at a horizontal speed of 1.5 m/s. Each integration of the ode23 function results in a new X and Z position of the REMUS vehicle. These positions are saved and used for plotting purposes.

B. ALTIMETER AND ALTITUDE CONTROLLER

Prior to designing an obstacle avoidance algorithm, the REMUS altitude controller must be created and tested. The altitude controller receives feedback from the altimeter and typically maintains an altitude of 3m above the ocean floor during searching operations. In the remusderivalt.m function, the altimeter is simulated by comparing the Z position of the REMUS to the time-domain Z_t array. It can therefore be seen that the Z_t array was required to be in the time-domain so that the REMUS position could be matched the corresponding segment of the ocean floor. The difference between the Z position of the REMUS, or its depth, and Z_t is the altitude.

The remusderivalt.m function will now track an altitude that can vary by either the depth change of the REMUS or by a change in the ocean floor depth. The altitude controller is created in this function is actually a modified depth controller, used for ease of manipulation with know depth-based equations of motion. In practice, it is the altitude that is sensed and compared with the altitude command, 'altcom'. The remusderivalt.m function actually uses a depth error in its feedback controller, but the command is received by subtracting the command altitude from the known depth, such that:

$$depthcom = H - altcom \tag{34}$$

Figure 10 illustrates a successful run of the altitude controller from an initial altitude of 6m down to its commanded altitude of 3m. This controller responds with no overshoot and minimal elevator plane action. This response was created by placing four poles at 0, -0.6, -0.62 and -0.63 and $\varphi = 0.1$.

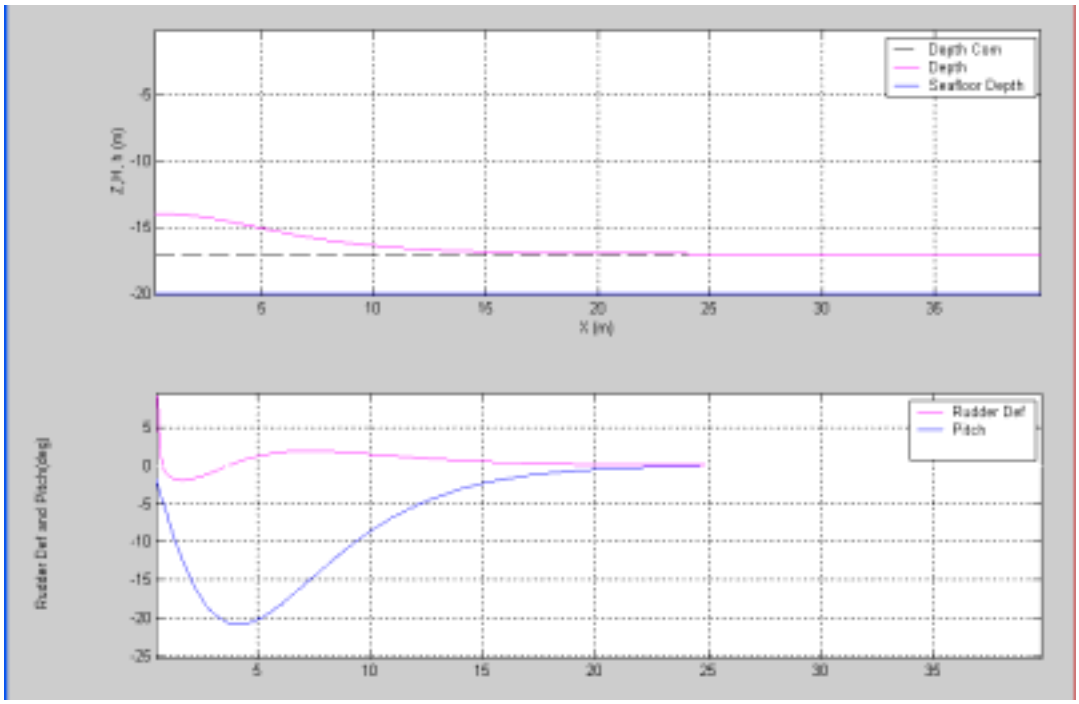

Figure 10. Altitude Controller at constant ocean floor depth as simulated in the REMUS model

C. SONAR MODEL

Following the successful test of the altitude controller, the forward-look sonar model yielding bearing and range to obstacles must be created in the MATLAB function. In real-world analysis the forward-look sonar will have much scatter and interference that may make this bearing and range information difficult to obtain. For purposes of this test, a forward-look sonar has been created without accounting for the scatter and interference that will have to be dealt with. The forward-look sonar model in remusderivalt.m is simply the trigonometrically determined values of bearing/range from the know position of the REMUS vehicle, its X and Z position, to values in the space-domain arrays of X_Model and Z_Model. As the ode23 function integrates each time step, the sonar model is used to compute the distance from the REMUS to every point in the X_Model and Z_Model array that is forward of the REMUS' X position. A series of if/then statements then filter out measurements that would be beyond the scope of the forward-look sonar. Bearings are based on a zero degree reference line parallel to the ship's x-

axis and starting at the ship's nose. This bearing is based on the REMUS' local body axis and therefore the pitch of the vehicle is accounted for as well, an example is shown below in Fig. 11.

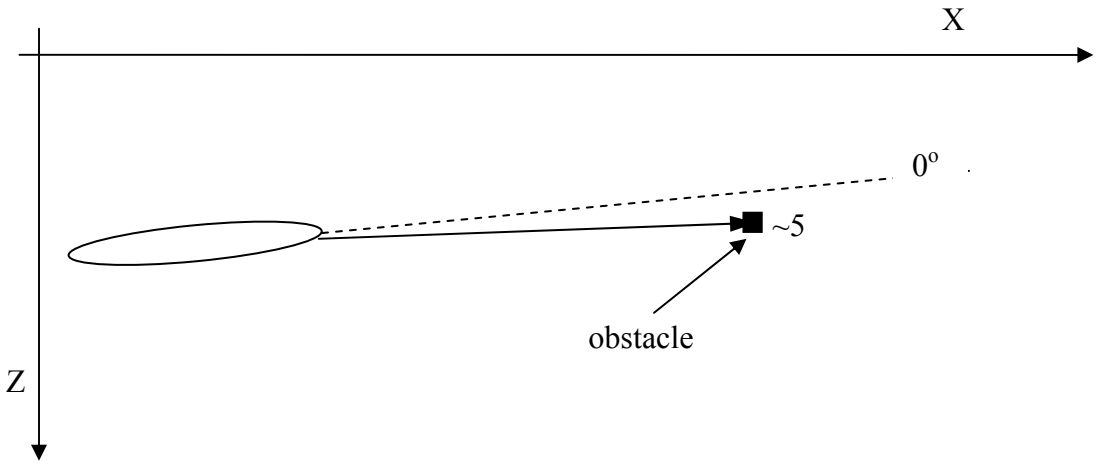

Figure 11. Depiction of Sonar Bearing with vehicle pitched in the Global X-Z Plane

The forward-look sonar should also be able to determine the height of the obstacle. To simulate this ability a variable, named HEIGHT, is created. The height of obstacle is determined by a loop that measures the difference in Z_Model for each successive position in the array. Once the difference equals zero it is assumed that the height of the object has been found and the variable HEIGHT is set to equal the first value in the Z_Model array whose difference from the next position in the array is zero (see Fig. 12). It should be stressed that this simulation works only for the purpose of this MATLAB function and other types of analysis will be required for actual forward-look sonar.

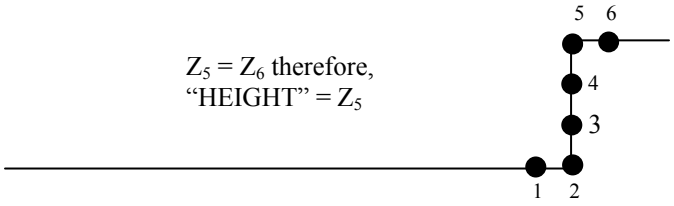

Figure 12. Illustrations depicting Sonar Model's determination of obstacle height

D. OBSTACLE AVOIDANCE ALGORITHM

With a working sonar model created in remusderivalt.m, logic code was developed to simulate how the REMUS will avoid obstacles. This logic includes the zone which triggers obstacle avoidance if an object enters, the sloping altitude command and the time delay that accounts for the blind spot between the forward-look sonar and the altimeter.

The obstacle avoidance algorithm, remusderivalt.m was tested on two different types of obstacles. One obstacle (obstacle A) simulates some type of sudden obstruction, perhaps a seawall, which rises up sharply and then returns to the previous ocean depth. The second obstacle (obstacle B) also rises abruptly, but maintains its height, similar to a coral reef. Examples of these obstacles are shown below in Fig. 13.

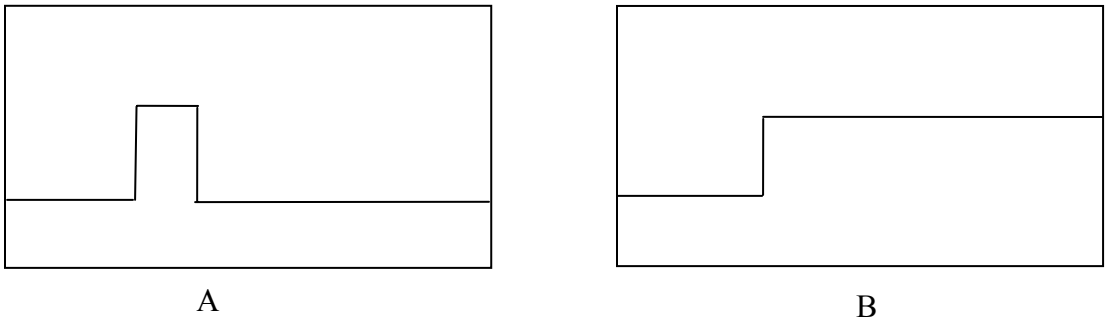

Figure 13. Obstacles used in testing avoidance algorithm

1. Obstacle Avoidance Zone

The forward-look sonar simulation is further refined by only examining objects that lay within the zone for obstacle avoidance. This zone is only 35 m in range and has a 25 degree view in the vertical plane. The sonar simulation examines each position of the ocean floor using the X_Model and Z_Model array. If the range and bearing from the REMUS position to the ocean floor is within the model a global variable, named TRUE is set to one. Once TRUE = 1 the obstacle avoidance portion of the function begins. If there is no object within the zone then TRUE remains equal to zero and the REMUS continues on its normal path with only the altitude controller manipulating the depth of REMUS as necessary.

2. Initial Tests

The final obstacle avoidance algorithm was not developed immediately. It was a process of trial and error, eliminating concepts and developing new ones based on old results. Initially an added pitch command was used to avoid an obstacle. Once an obstacle was in the field of view of the forward-look sonar, REMUS received a pitch command that increased the altitude. However, results showed that the pitch command and the altitude controller battled each, other and as commanded pitch was obtained, the altitude controller began forcing REMUS back down to its original altitude. The result was a difficult to predict flight path that had to be tailored for different types of obstacles. Additionally, the high angles of pitch resulting from this method typically caused the obstacle to move into the blind spot. Once the obstacle was in the blind spot REMUS pitched downwards until it regained the obstacle and then pitched upward. This created a sinusoidal flight path that was both inefficient and unsafe. Fig. 14 displays an initial test of an obstacle avoidance algorithm using pitch command. This figure demonstrates the sinusoidal flight path and the battle between the pitch command and the altitude command.

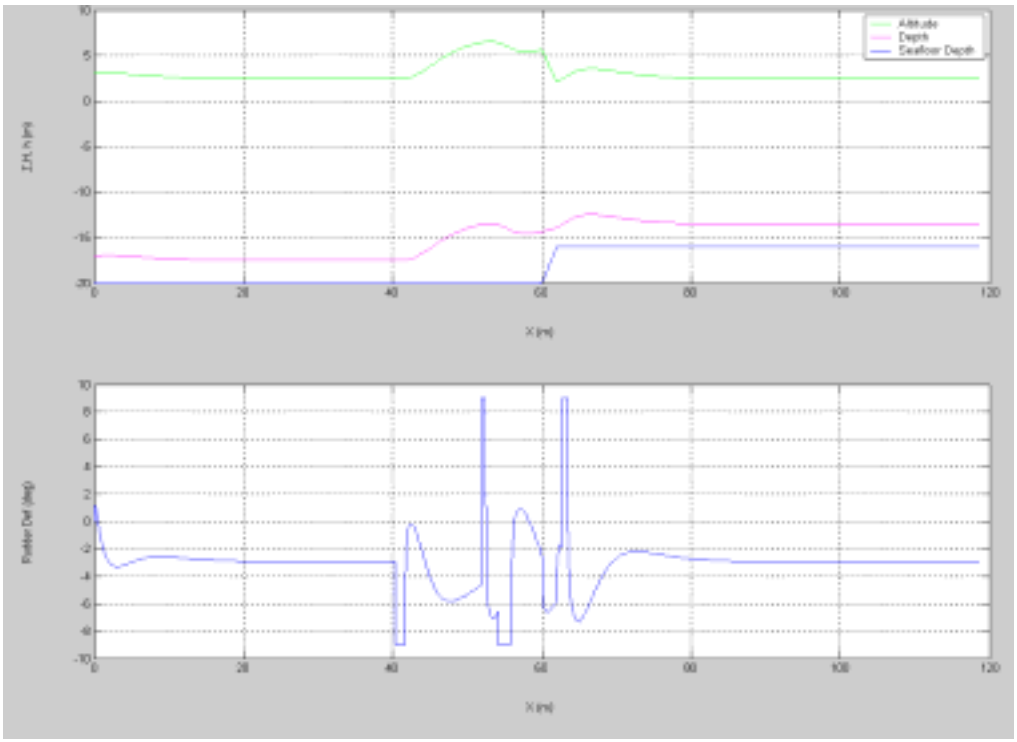

Figure 14. Initial Obstacle Avoidance Algorithm Results

A solution to the battle between the pitch and altitude command was developed by creating a new altitude command that drove the REMUS up to a new altitude in support of the pitch command. The altitude command was based on the height of the obstacle and was an on/off command. Now the REMUS received both a pitch command and an altitude command when an obstacle was located. The higher altitude command provided for a smoother flight above the obstacle but difficulties still occurred due to the blind spot. The REMUS was now experiencing strong commands in pitch and altitude as the obstacle went in and out of view causing a more erratic flight motion and wasteful elevator plane motion. Fig. 15 displays the effects of this pitch and altitude command controller. The bottom graph in Fig. 15 also displays the TRUE variable, scaled for graphical representation. Fig. 15 shows TRUE changing from zero to a non-zero value. This on/off value of TRUE shows when the obstacle is in or out of the blind spot. When the obstacle is in the blind spot, or has not been detected at all, TRUE is equal to zero. When the obstacle is being detected by the sonar TRUE is equal to one and is scaled to 20 for graphical purposes only.

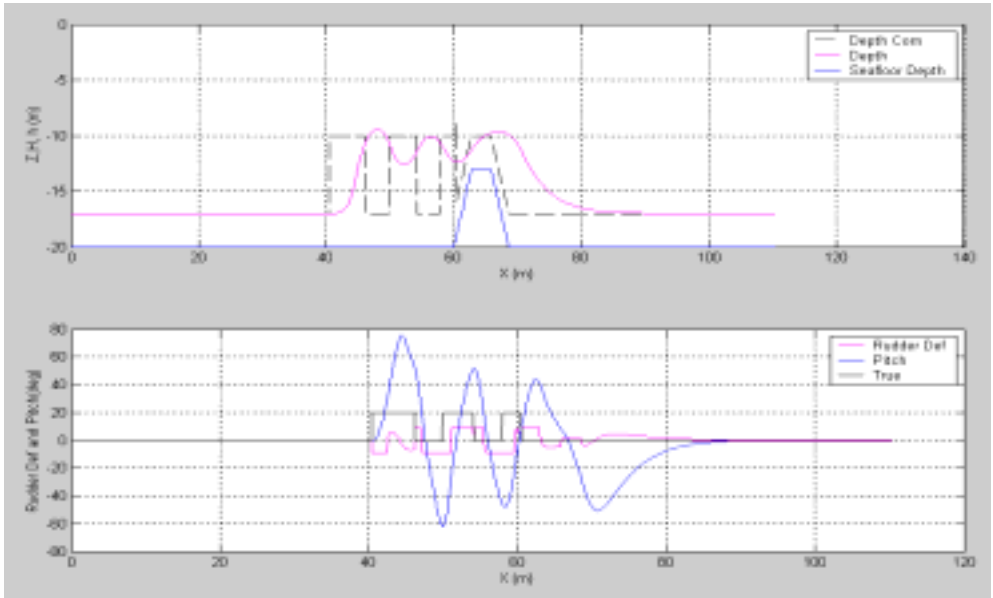

Figure 15. Obstacle Avoidance Results using pitch and altitude command

3. Advanced Tests

Initial tests of the altitude controller made it clear that it would be possible to see an obstacle and maneuver to avoid it. However, problems with the blind spot between

the altimeter and the forward-look sonar caused erratic flight paths that were difficult to predict and wasted important energy in elevator plane motion. To smooth out the flight path and improve its predictability the sloping altitude command was developed.

The sloping altitude command was developed as follows: The horizontal position of REMUS when the object is first detected is set as the variable SSTART. This position anchors the sloping altitude equation. The sloping altitude equation is a simple linear equation in the form of:

$$y = mx + b \tag{35}$$

where m is the slope which determined from the quotient of the height of the obstacle and the range to the obstacle and b = SSTART. The slope, m, is altered slightly so that the altitude command reaches the height of the obstacle some 10 m in front of the obstacle. Fig. 16 illustrates the geometry of the altitude command.

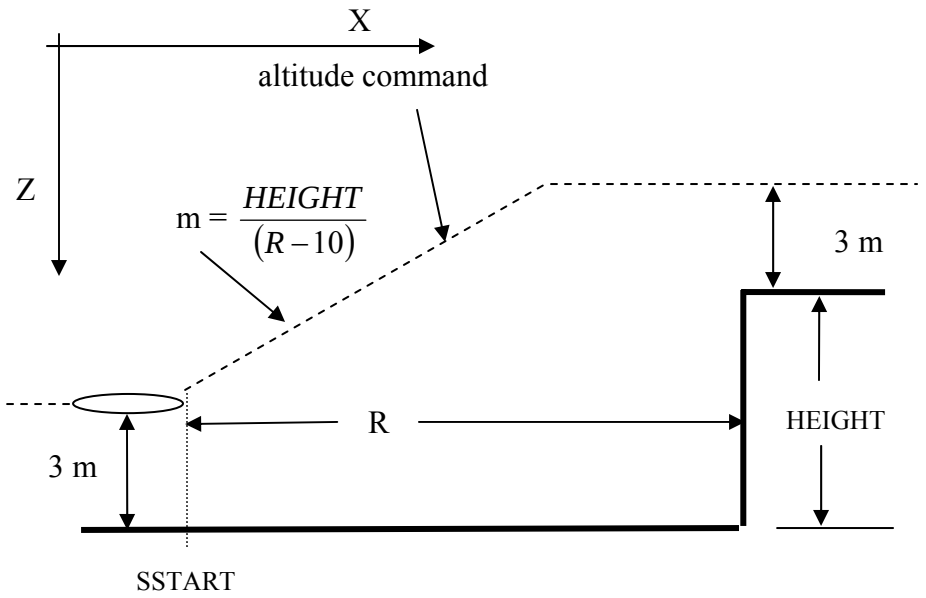

Figure 16. Sloping Altitude Command Generator for Obstacle Avoidance

Fig. 17 illustrates the first attempts at this type of sloping altitude control. The sloping altitude control still faced problems due to the blind spot. Fig. 17 illustrates this

clearly as the altitude command can be seen to “ratchet” from the sloping line back down to the original altitude every time the obstacle entered the blind spot.

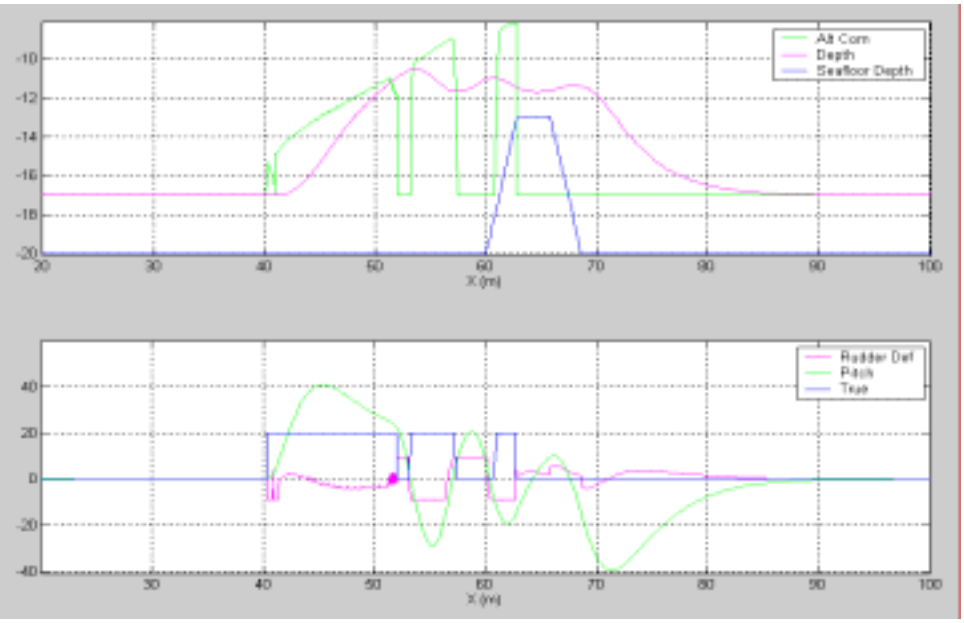

Figure 17. Obstacle Avoidance Results using a Sloping Altitude Command

There are two methods used to solve the problem of the blind spot between the altimeter and the forward-look sonar. The first and most direct method was to limit the pitch of REMUS as much as possible. Although pitch command seemed necessary at the beginning of the development of the obstacle avoidance algorithm, it seemed less necessary as the altitude controller was developed. By setting the pitch command to zero at all times forced REMUS to maintain a very shallow pitch and return to zero as soon as the proper altitude was gained. Fig. 18 shows the flight path of the REMUS with a sloping altitude command and a zero angle pitch command. Under the proper conditions, this is all that is necessary to successfully avoid an obstacle in the vertical plane. The obstacle is always within the zone of the forward-look sonar and therefore a smooth flight path is created by the sloping altitude command.

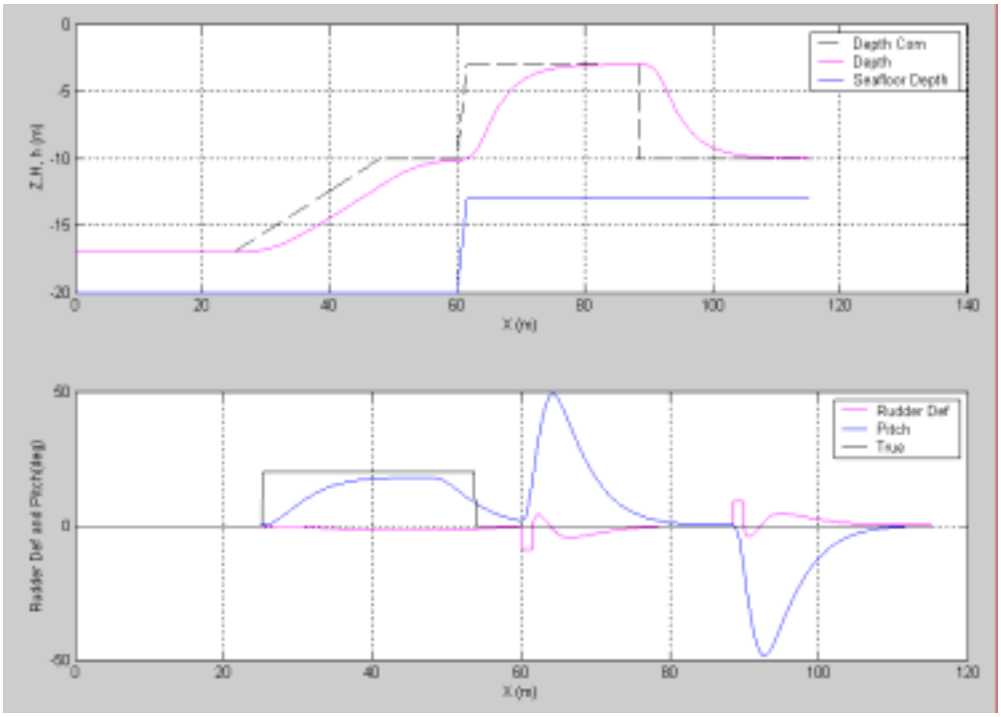

Figure 18. Obstacle Avoidance Results using Sloping Altitude Command and Zero Pitch Command

The possibility does exist however, that an obstacle may still enter the blind spot. Most likely this would be due to a failed detection of an obstacle due to sonar interference or perhaps because REMUS turns into the obstacle and the range is less than 35m. In this case REMUS would pitch higher to ascend quickly and could lose sight of the obstacle. To account for this possibility a delay was created. This delay prevents REMUS from pitching down immediately after losing sight of an obstacle.

The delay is created as follows: At the same time that TRUE is set to one, another global variable, named DDIST is set to equal the range to the obstacle plus the horizontal position of the REMUS. This variable creates the time delay required to account for the blind spot between the altimeter and the forward-look sonar. Without this delay, any time the obstacle is removed from the field of view of the forward-look sonar REMUS begins to pitch downwards to original altitude, even if the object is only a few meters away. By creating an if/then statement in remusderivalt.m, the obstacle avoidance altitude command remains in effect unless both TRUE = 0, indicating that the obstacle is

no longer in the zone, and that REMUS is beyond the horizontal position of DDIST, indicating that REMUS has passed the estimated position of the object.

4. Final Results

The final obstacle avoidance algorithm uses the sloping altitude command as well as the delay to account for the blind spot. A final problem arose due to an overlap in altitude command and the altimeter signal of ocean floor depth. As REMUS rises above an obstacle it receives a command to fly to certain height above the ocean floor. However, once the ocean floor rises to the height of the obstacle there can be an overlap where the REMUS is actually being commanded to fly at that same height above the obstacle. For example, REMUS may be attempting to fly to 10 m above the ocean floor to pass over a 7 m obstacle. Once REMUS is directly above the obstacle the overlap will cause REMUS to attempt to fly 10m above the obstacle instead of the 3m altitude desired. A trigger required to be developed that would either stop the sloping altitude command or limit the sloping altitude command once REMUS above the obstacle. In remusderivalt.m a simple if/then statement accomplished this by setting the altitude command back to 3 m once the ocean floor depth equaled the calculated depth based upon the forward-look sonar's determination of the obstacle's height. The final version of the obstacle avoidance algorithm is shown in Fig. 19. The flight path is a gradual increase in altitude with a minimum amount of rudder used. It can be seen that even after the TRUE variable returns to zero that the elevator plane does not jar in response. Instead the elevator maintains its gradual return to it neutral position.

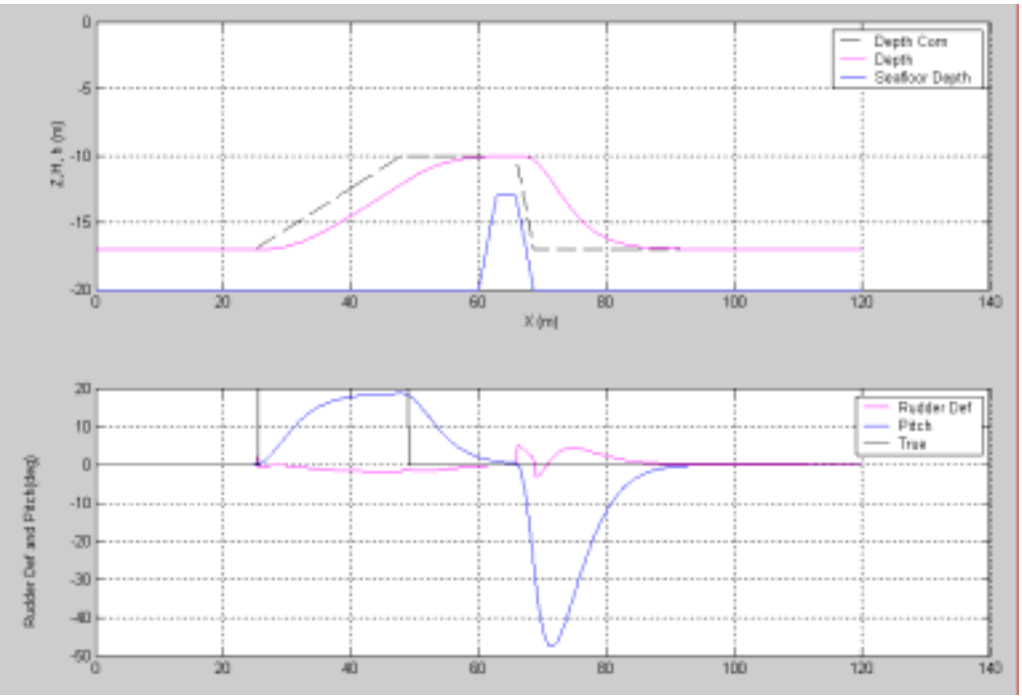

Figure 19. Final Obstacle Avoidance Results for Obstacle A

Additionally, remusderivalt.m was tested for obstacle B, with the results shown below in Fig. 20. The same characteristics demonstrated for obstacle A were exhibited in obstacle B.

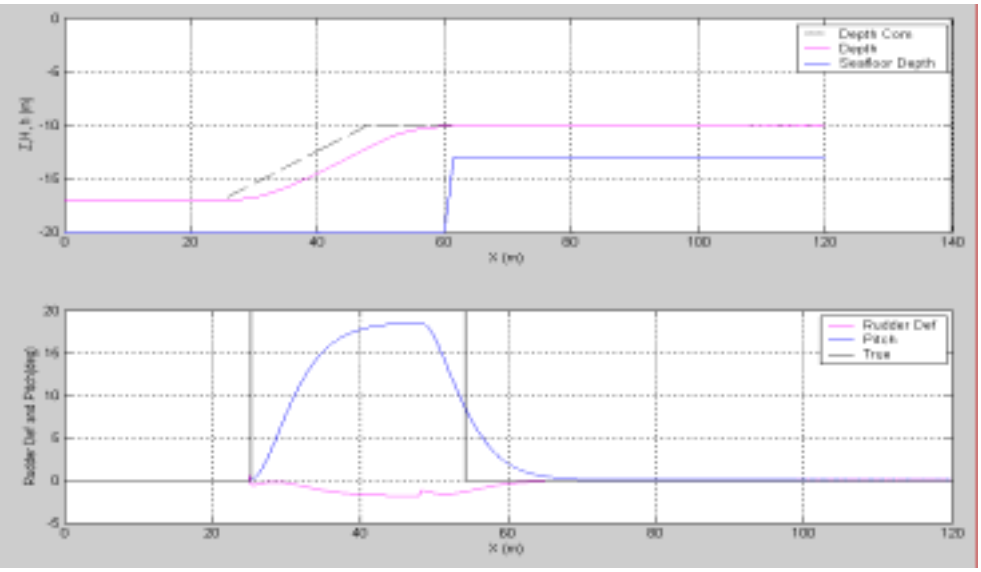

Figure 20. Final Obstacle Avoidance Results for Obstacle B

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSION

The results from these tests indicate only one of many possible solutions to obstacle avoidance in the vertical plane. Unlike horizontal obstacle avoidance examined by others, vertical obstacle avoidance does not have a track to follow. The vertical plane is strictly a reactionary environment. Altitude is maintained by a feedback controller and altimeter input for most gradual changes in elevation. Once a significant obstacle has been detected that must be avoided by ascending over it a challenge exists to command the AUV to rise while still dealing with the altitude controller that is trying to maintain the AUV at its present depth. While one option may be to "turn off" the altitude controller until the AUV has passed the obstacle, this thesis examines the possibility of using the altitude controller to avoid the obstacle. By creating a sloping altitude command that causes the AUV to rise above the obstacle no sensors need be turned off or ignored. This seems fundamental to this problem mainly due to the fact that there may be more obstacles beyond the first obstacle detected.

After trying numerous methods to avoid a vertical obstacle it seems best to maintain a small pitch angle, and a gradually increasing altitude command. The problems faced by having a blind spot between the AUV's forward-look sonar and its altimeter create many problems otherwise due the constant decision making process that the AUV is capable of. A simple reactive based controller provides the AUV with the most flexible and adaptive capability. Two different obstacle types were studied and successfully avoided using the proposed algorithm.

B. RECOMMENDATION

There are many paths of future research in this area of obstacle avoidance. These options are due to the simplifications and assumptions made to accomplish this study. It is obvious that a next eventual step will be the creation of an obstacle avoidance algorithm designed for use in a three dimensional domain. Work at this institute has already been done with obstacle avoidance in the horizontal plane (Fodrea). In that case,

as well as in this case, derivation of the equations of motion was simplified by assuming that certain motions were zero. For example, there is no roll in the calculations made here concerning vertical obstacle avoidance. Similarly, there are no pitch concerns with the work done concerning horizontal obstacle avoidance. In the 3-D environment, the equations of motion will become highly non-linear once all motion is considered.

Another complexity involved in the 3-D environment will involve the logic of determining what type of an obstacle is in front of the AUV. In this case of research and in Fodrea's it was know that the vehicle was going to avoid it either vertically or horizontally. In the 3-D world how will an AUV determine its best course of action? If an AUV begins to turn left to avoid an obstacle and the obstacle is a 50ft long sea wall, it should have ascended. On the other hand the AUV could try to rise over a 30 ft column that it could have easily gone around.

There is still much work; however, that can be done concerning the study of obstacle avoidance in the vertical plane alone. This study has not examined the impact of multiple obstacles for example. In particular, a stepped obstacle, where an AUV may have to rise to one elevation and then quickly to another needs to be examined. This type of obstacle can be quite common in the littorals, particularly in reef formation. Another type of obstacle can be called the "sudden obstacle." This sudden obstacle is an obstacle that is not detected until the range from it is very short. This type of obstacle could occur by turning into it or could result from poor sonar data. In cases of sudden obstacles, high pitch angles may result, placing the obstacle into the AUV's blind spot. It may also be simply not possible to avoid a sudden obstacle by pitch upwards. A combination of speed changes and altitude command may be necessary to avoid some sudden obstacle.

This thesis also assumes that all sonar detection will be dead on accurate. Real world interference problems, such as background noise scatter and bottom bounce, make this type of detection extremely unlikely. Work could also be done to attempt to model a more realistic form of sonar. As work in forward-look sonar develops, actual sonar data would be an excellent way to examine this area.

APPENDIX I: MATLAB CODES

remus.run

```
clear
clc
z_g = 1.96e-2;
x_b = 0;
W = 299;
buoy = 306;
I_z = 3.45;
I_y = 3.45;
I_x = 1.77e-1;
U = 1.5;
to = 0;
tf = 80;

global TRUE;
global DDIST;
global HEIGHT;

TRUE = 0;
DDIST = 0;
HEIGHT = 0;
m = 299/9.81;
M_q = -6.87;
M_qdot = -4.88;
M_w = 30.7;
M_wdot = -1.93;
M_d = -34.6;
Z_q = -9.67;
Z_qdot = -1.93;
Z_w = -66.6;
Z_wdot = -35.5;
Z_d = -50.6;

% Dynamics ------------------------------------------------------------
M = [m-Z_wdot -Z_qdot 0 0;-M_wdot I_y-M_qdot 0 0;0 0 1 0;0 0 0 1];
A_0 = [Z_w m*U+Z_q 0 0;M_w M_q -z_g*W 0;0 1 0 0;1 0 -U 0];
B_0 = [Z_d;M_d;0;0];

A = inv(M)*A_0;
B = inv(M)*B_0;
C = [0 0 0 1];
D = inv(M)*[0;0;0;0];

% Pole Placement ------------------------------------------------------
-
p = [0 -0.6 -0.62 -0.63];
k = place(A,B,p);

Ac = A-B*k;
[V,v] = eig(Ac');
s = V(:,4);
```

```
% Controller ------------------------------------------------------
---
x0 = [0;0;0;17;0;U;s;k';[0;0;0;3]];                              % initial
condition and command

[t,x] = ode45(@remusderivalt,[to tf],x0);

TRUE = 0;
DDIST = 0;
HEIGHT = 0;

for i = 1:length(t)
    [xdot,ds,sig,sigdot,h,TRUE,depthcom]=remusderivalt(t(i),x(i,:)');
    T(i) = TRUE;
    DEP(i) = -depthcom;
    sigma(i) = sig;
    alt(i) = h;
    deltasp(i) = ds*180/pi;
    H(i) = alt(i) + x(i,4);
end;

% Plotting ---------------------------------------------------------
---------

     subplot(2,1,1),plot(x(:,5),DEP,'k--',x(:,5),-x(:,4),'m',x(:,5),-
H,'b'),grid

subplot(2,1,2),plot(x(:,5),deltasp,'m',x(:,5),x(:,3).*180/pi,x(:,5),T*2
0,'k'),grid


     subplot(2,1,1),xlabel('X (m)')
     subplot(2,1,1),ylabel(' Z,H, h (m)')
     subplot(2,1,1),legend('Depth Com','Depth','Seafloor Depth')
     subplot(2,1,2),xlabel('X (m)')
     subplot(2,1,2),ylabel('Rudder Def and Pitch(deg)')
     subplot(2,1,2),legend('Rudder Def','Pitch','True')
     subplot(2,1,1),axis([0 140 -20 0])
```

remusderivalt.m

```
function[xdot,ds,sig,sigdot,h,TRUE,depthcom] = remusderivalt(t,xx);
%
%
% remusderivalt is an smc controller that is called up by an
% ode function commanding the vehicle to a specific altitude.
% Created by Chris Chuhran, May 1, 2003

% REMUS parameters -------------------------------------------------

U = xx(6);
s = xx(7:10);
k = xx(11:14)';
```

```
x = xx(1:4);        % x(1) = q, x(2) = w, x(3) = theta, x(4) = Z, xx(5)
= X
xcom = xx(15:18);   % xcom = [q com, w com, theta com, depth com]
z_g = 1.96e-2;
x_b = 0;
W = 299;
buoy = 306;

global TRUE;
global DDIST;
global HEIGHT;

I_z = 3.45;
I_y = 3.45;
I_x = 1.77e-1;
U = 1.5;
m = 299/9.81;
M_q = -6.87;
M_qdot = -4.88;
M_w = 30.7;
M_wdot = -1.93;
M_d = -34.6;
Z_q = -9.67;
Z_qdot = -1.93;
Z_w = -66.6;
Z_wdot = -35.5;
Z_d = -50.6;


thetacom = 0;
altcom = 3;

R = 35;            % Sonar Range (m)
SSTART = 60 - R;    % this variable needs to be named once for each
obstacle, hardwired for now
% Dynamics ----------------------------------------------------------
-
M = [m-Z_wdot -Z_qdot 0 0;-M_wdot I_y-M_qdot 0 0;0 0 1 0;0 0 0 1];
A_0 = [Z_w m*U+Z_q 0 0 ;M_w M_q -z_g*W 0;0 1 0 0;1 0 -U 0];
B_0 = [Z_d;M_d;0;0];

A = inv(M)*A_0;
B = inv(M)*B_0;
C = [0 0 0 1];
D = inv(M)*[0;0;0;0];

% Seafloor Modeling for Sonar (non-time dependent) -------------------
---
% Seabottom I -------------------------------------------------------
---
% X_1 = [0:0.5:60];
% X_2 = [60:0.2:62.8];
% X_3 = [62.8:0.5:65.8];
% X_4 = [65.8:0.2:68.6];
% X_5 = [68.6:0.5:111.6];
% X_Model = [X_1 X_2 X_3 X_4 X_5];
%
```

```
% H_1 = 20*ones(1,121);
% H_2 = [20:-0.5:13];
% H_3 = 13*ones(1,7);
% H_4 = [13:0.5:20];
% H_5 = 20*ones(1,87);
% H_Model = [H_1 H_2 H_3 H_4 H_5];

% Seabottom II ------------------------------------------------------
---
X_1 = [0:0.5:60];
% X_2 = [60:0.25:62];
% X_3 = [62:0.5:121];
X_2 = [60:0.1:61.4];
X_3 = [61.4:0.5:121.4];
X_Model = [X_1 X_2 X_3];

H_1 = 20*ones(1,121);
% H_2 = [20:-0.5:16];
% H_3 = 16*ones(1,119);
H_2 = [20:-0.5:13];
H_3 = 13*ones(1,121);
H_Model = [H_1 H_2 H_3];


% Sonar -------------------------------------------------------------
--
for d = 1:length(X_Model)
    if X_Model(d) > xx(5)
        range = sqrt((X_Model(d) - xx(5))^2 + (H_Model(d)-x(4))^2);
        bearing = asin((H_Model(d) - x(4))/range) + x(3);          %
bearing to object as read by sonar (pitch corrected)
        floor_brng = asin(2.4/R) + x(3);         % this is bearing when
ocean floor is 'R'm away

        if (bearing - x(3)) == 0           % prevents divide by zero
error (sin(angle))
            floor_alt = 100;              % this happens when obstacle
is directly in front of REMUS
        else
            floor_alt = 2.4/sin(bearing - x(3));     % this is range to
ocean floor minus buffer
        end                                             % buffer of 0.6
can be handled by altitude control

        if (bearing > 0 & bearing < floor_brng & range < R) | (bearing
> floor_brng & bearing < 12*pi/180 & range < floor_alt)

            TRUE = 1;
            DDIST = range + xx(5) + 0;      % ensures no dive before
obstacle is passed

            for dd = d:length(X_Model)
                if abs(H_Model(dd) - H_Model(dd-1)) <= 0.001
                    HEIGHT = 20 - H_Model(dd);
                    break
                end
            end
```

```
            break

        else TRUE = 0;
        end
    end
end

% Controller ------------------------------------------------------
--
if ((TRUE == 1)| (xx(5) < DDIST))

    altcom = 3 + (xx(5) - SSTART)*HEIGHT/(R-12);

    if altcom > HEIGHT + 3
       altcom = 3 + HEIGHT;
    elseif altcom < 3
       altcom = 3;
    end
end;

% Seafloor Modeling for Controller (time dependent)
% Seabottom I -----------------------------------------------------
---
% if xx(5) <= 60 | xx(5) >= 68.6
% H = 20;
% elseif (xx(5) > 60 & xx(5) <= 62.8)
%     H = 170 - 2.5*xx(5);
% elseif xx(5) > 62.8 & xx(5) <= 65.8
%     H = 13;
% elseif xx(5) >65.8 & xx(5) < 68.6
%     H = -151.5 + 2.5*xx(5);
% end

% Seabottom II ----------------------------------------------------
----
if xx(5) <= 60
    H = 20;
elseif (xx(5) > 60 & xx(5) <=61.4)
% elseif (xx(5) > 60 & xx(5) <=62)
%     H = 140 - 2*xx(5);        % depth = 16m
    H = 320 - 5*xx(5);        % depth = 13m
elseif xx(5) > 61.4
    H = 13;
end

depthcom = H - altcom;             % altitude control must be converted
to depth control for EOM

if ((TRUE == 1) | (xx(5) < DDIST))          % prevents jump up at edge
    depthcom = 20 - altcom;                   % hardwired for now, need
to "look back"
end

xcom=[0;0;thetacom;depthcom];
phi = 0.1;
sig=s'*(x-xcom);
```

```
Nmax= 2;
ada = Nmax*0.4/inv((s'*B));
delta = -k*x-Nmax*0.4*sign(inv((s'*B)))*tanh((sig/phi));


if abs(delta) > 0.157                          % REMUS has nine deg max
rudder deflection
    delta = 0.157*sign(delta);
end

h = H - x(4);                                      % depth for plotting
purposes
ds = delta;                                    % rudder angle for plotting
purposes
xsdot = A*x+B*ds+D;
sigdot = s'*xsdot;
xsdot(4) = [x(1)*cos(x(3))-U*sin(x(3))];   % Large angle approximation
xxdot = [U*cos(x(3))+x(1)*sin(x(3))];      % Horizontal advance
xdot = [xsdot;xxdot;0;0;0;0;0;0;0;0;0;0;0;0;0];
```

LIST OF REFERENCES

Bennet, Andrew A., et al.,"Bottom Following for Survey-Class Autonomous Underwater Vehicles", Ninth International Symposium on Unmanned Untethered Submersible Technology, September 1995 pp.327-336.

Blidberg, Richard D., "The Development of AUVs: A Brief Summary", Autonomous Undersea Systems Institute, ICRA, Seoul, Korea, May 2001.

Clark, Vernon, "Seapower 21, Projecting Decisive Force Capabilities", United States Naval Institute Proceedings, October 2002, www.usni.org .

Fodrea, Lynn, "Obstacle Avoidance Control for the REMUS Autonomous Underwater Vehicle", Naval Postgraduate School, December 2002.

Healey, A. J., *Dynamics of Marine Vehicles (MA-4823),* Class Notes, Naval Postgraduate School, Monterey, CA, 1995.

Healey, A. J., "Command and Control Demonstrations with Cooperating Vehicles", ONR Research Proposal in response to ONR BAA 01-012 "Demonstration of Undersea Autonomous Operation Capabilities and Related Technology Development", August 2001.

Kamon, I and Rivlin, E., "Sensory-based motion planning with global proofs", IEEE Transaction on Robotics and Automation, Vol 13, no. 6, 1997.

Singh, Hanumant, et al., "Sonar Mapping with the Autonomous Benthic Explorer (ABE), Ninth International Symposium on Unmanned Untethered Submersible Technology, September 1995 pp. 367-375.

Stutz, Douglas, "UUVs in Iraq", Navy News, story # NNS030829-05, 8/29/2003.

Thompson, R.L., "Two Dimensional and Three Dimensional Imaging Results Using Blazed Arrays", OCEANS, 2001. MTS/IEEE Conference and Exhibition Vol. 2, November 2001, pp. 985-988.

von Alt, Christopher, et al., "Remote Environmental Measuring Units", Proceedings of the 1994 Symposium on AUV Technology, July 1994, pp. 13 – 19.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Naval/Mechanical Engineering Curriculum Code 34
   Naval Postgraduate School
   Monterey, CA

4. Professor Anthony J. Healey, Code ME/HY
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA

5. Dr. Donald Brutzman, Code UW/Br
   Undersea Warfare Group
   Naval Postgraduate School
   Monterey, CA

6. Dr. T. Swean, Code 32OE
   Office of Naval Research
   Arlington, VA

7. Doug Horner
   Naval Postgraduate School
   Monterey, CA

8. Christopher J. Von Alt
   Woods Hole Oceanographic Institution
   Woods Hole, MA

9. LT Chris Chuhran
   Puget Sound Naval Shipyard
   Bremerton, WA